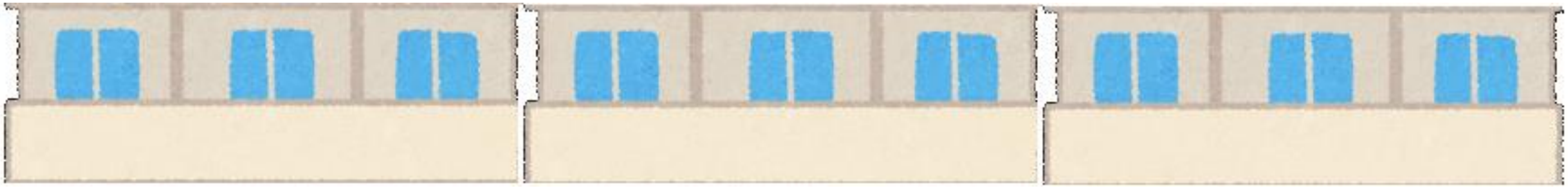
細長い屋敷(Long Mansion)

hogloid



問題概要

- 部屋がN個直線形につながっており、隣接する部屋への移動には鍵が1つ必要
- 部屋には鍵が落ちてる。一度拾った鍵は何度でも使える
- クエリをQ個処理:ある部屋からある部屋へ行けるか？
- N,Q,鍵の総和(=M) <= 10^5
- 小課題3:鍵の種類(=K) <= 20

小課題1

- クエリごとに独立に処理
- ある点から進み到達できる範囲は必ず区間になる、区間を求めるには開始地点のみを含む区間を左右に拡張し鍵を回収し続けることで求まる
  - 鍵を取ることで損することはないため
- クエリの目的地が始点から行ける区間に含まれるか求める。O(Q(N+M))

小課題2

- N個の始点について到達可能区間を予め求めておく。クエリーにはO(1)で答える。
- O(Q+NM)

小課題3

- N個の始点について到達可能区間を予め求めておく。クエリーにはO(1)で答える。
- 始点から右に向かって初めてある鍵が必要になる廊下のみが区間の右端になりうる
- 始点から右に向かって初めてある鍵が手に入る部屋でのみ回収することに価値がある
  - 左についても同様。見る必要のある部屋は高々O(K)
- これで座標圧縮をすると始点ごとにO(KlogK)

満点(その1)

- iから到達できる区間を$[l_i,r_i]$とする
- iからjに到達できるなら、$[l_i,r_i]$は$[l_j,r_j]$を含む・・①
- これを使って区間を左右に拡張するメモ化再帰ができるのでは？
  - 単純にやると、できない
  - 互いに到達可能な始点があると遷移にサイクルがあり終了
- しかし、①より互いに到達可能なら区間は等しい

満点(その1)

- よって、無限ループに陥りそうになるとき、その遷移のサイクルに属する始点全ての区間は等しい
- メモ化再帰を行うと遷移はdfs木になるので、無限ループが起こりそうになる遷移は後退辺で、サイクルに属する始点はdfs木の現在の始点の祖先
- サイクルで最も祖先の始点に戻るまで、区間を併合しながらdfsをやめてしまえばよい。dfsをやめてしまった始点はUnionFindで「どの始点と答えが同じか」を記憶しておく

満点(その1)

- 計算量は？
  - 全ての始点について、併合によらない区間の拡張回数が全体の計算量
  - 右端をkからk+1に拡張する始点i, j(i<j<=k)があったとする
    - このとき、iは右端をkからk+1に拡張する前にjについて再帰的に問題を解いており、始点jからの区間を含むため、矛盾
  - よって、各廊下について左から拡張される回数は1回、右からについても同様
  - 区間の拡張可能性判定(=鍵を持っているかどうか)にO(logN)かかる。UnionFindも使うため、合計O(NlogN)

満点(その2)

- 全ての始点iについて、はじめ[i,i]から始め、右・左に拡張していくが、1ステップでそれ以上右に拡張できなくなるまで拡張する
  - 右への拡張を考える。すべての廊下について、それより左で最も右の廊下を通る鍵が落ちている部屋の場所を覚えておく
  - 右へ拡張できなくなるところは、初めてその廊下を通るのに必要な鍵が、現在の区間より左にある場所
    - これは二分探索segment treeを使えばいけます

満点(その2)

- 毎ステップに左右へ拡張し、答えを求める
  - TLEします(真ん中にN/2個互いに通行可能な始点があり、その周りをジグザグ動くようなケース)
- 区間の経過全てについて、答えをメモしておき、既に求めた区間を訪れたらメモした答えを返すと、状態数はO(N)になる！

満点(その2)

- それ以上拡張できない区間を終区間と呼ぶ
- 終区間・長さ1の区間はO(N)個しかないので、それ以外を数える
- 左端がlで、これから右に拡張していく区間について考える
- $[l,r_1],[l,r_2],\ldots,[l,r_K]$とおく。$(r_1>r_2>\ldots)$
- $[l,r_i]$を右に拡張すると、$[l,s_i]$になるとする。
- $s_i<r_1$ なら$[l,s_i]$は終区間 ($[l,r_1]$は左に拡張できなかったので、$[l,s_i]$も左に拡張できない)
- $s_i>r_1$ なら $s_i=s_1$ ($[l,s_i]$は$[l,r_1]$を含んでいるので、右端がr_1に達した時点でどこまで拡張されるかは同じ)
- なので、ある左端lから遷移する非終区間は1つ。

満点(その他)

- ダブリング、segment tree、etc…
  - 時間が余ったら口頭で説明します

得点分布

得点分布